

Case C1.2: Flow over the NACA0012 airfoil

Giorgio Giangaspero*, Edwin van der Weide†, Magnus Svärd‡, Mark H. Carpenter§ and Ken Mattsson¶

I. Discretization, iterative method and hardware

See the appendices.

II. Case summary

We submit 5 sets of results: 1 for the inviscid subsonic case, 2 for the viscous case (with sharp and with rounded trailing edge), 2 for the transonic case (with and without shock capturing, see below for the detailed description of the shock capturing scheme).

For this test case, we generated a fine O-grid of 577×513 vertices using the hyperbolic grid generation capabilities of the commercial software Pointwise [1]. The farfield is located at 1000 chords, as requested. The trailing edge is sharp, unless stated otherwise. For the subsonic configurations, the vertex distribution on the airfoil is the same on pressure and suction side, as shown in fig. 1. Instead, for the transonic configuration, vertices are clustered on the suction side, in particular close to the shock region, fig. 2. Initial guesses are obtained via grid sequencing, where appropriate. The coarser grids are obtained by deleting every other grid line from the finer grid (regular coarsening).

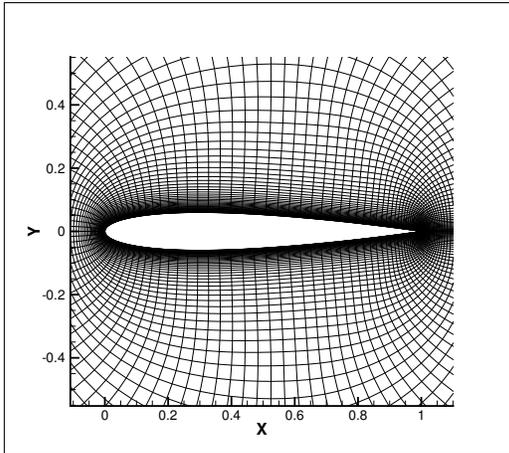


Figure 1. Coarse grid (145×129 vertices) for the subsonic configurations.

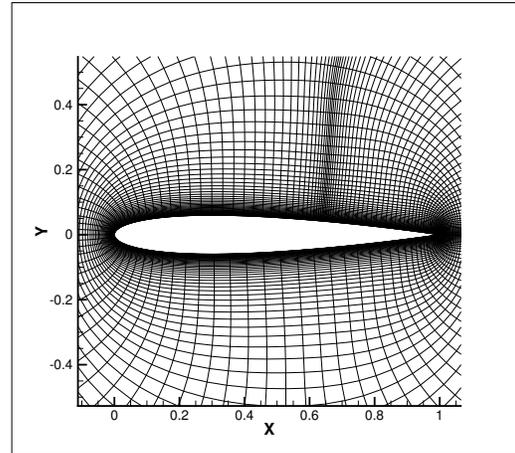


Figure 2. Coarse grid (145×129 vertices) for the transonic configuration.

Our 'converged' values of lift and drag coefficients are obtained by applying Richardson extrapolation to the 5th order results where possible. If the convergence history with the 5th order scheme is not monotonic (and therefore the Richardson extrapolation cannot be applied), we use the extrapolated value obtained with the 4th order scheme.

*Department of Mechanical Engineering, University of Twente, the Netherlands, e-mail: g.giangaspero@utwente.nl

†Department of Mechanical Engineering, University of Twente, the Netherlands, e-mail: e.t.a.vanderweide@utwente.nl

‡Department of Mathematics, University of Bergen, Norway, e-mail: Magnus.Svard@math.uib.no

§NASA Langley Research Center, Hampton, VA, e-mail: mark.h.carpenter@nasa.gov

¶Uppsala University, Uppsala, Sweden, e-mail: ken.mattsson@it.uu.se

Table 1. 5th order results for the subsonic inviscid configuration.

C_l	C_d	h	work units
2.86188101e-01	7.63623474e-06	7.31175e-03	11.0
2.86397817e-01	5.03047227e-06	3.66931e-03	52.3
2.86406774e-01	4.70338824e-06	1.83803e-03	373.5

II.A. Subsonic inviscid

Figure 3 shows the contours of the relative entropy error close to sharp trailing edge, which is the main source of entropy error. The reason is that the geometry is not smooth in this region, which results in the large entropy error originating from the trailing edge. Additional tests with a C-grid topology showed exactly the same behavior, hence the O-grid topology is not the cause of this error. As a consequence, the schemes do not obtain the design accuracy, at least not for the inviscid case. Our 5th order results are presented in tab. 1.

II.B. Transonic inviscid

For this configuration we computed two sets of results. One set obtained with 'standard' SBP-SAT artificial dissipation operators, given in [2]. In 1D, given p order of accuracy at the boundary, these operators have the form

$$\tilde{H}_p^{-1}(\tilde{D}_p^T(\alpha\Lambda)B_p\tilde{D}_p)u$$

where $\tilde{H} = H/h$ and H is the norm, Λ is a diagonal matrix with the spectral radius of the inviscid flux Jacobian, \tilde{D}_p/h^p is a first-order approximation of $\partial^p/\partial x^p$ (note that \tilde{D}_p is an undivided difference), $B_p + B_p^T \geq 0$, α is a non-negative constant and u represents the solution vector. The second set of results are obtained with the modified artificial dissipation operator:

$$\tilde{H}_p^{-1}(\tilde{D}_1^T(c_2\epsilon_2\Lambda)B_1\tilde{D}_1 + \tilde{D}_p^T(\alpha^*\Lambda)B_p\tilde{D}_p)u \quad (1)$$

The ϵ_2 term is the pressure-switch term firstly introduced in [3], i.e.

$$\phi_i = \frac{|p_{i-1} - 2p_i + p_{i+1}|}{|p_{i-1} + 2p_i + p_{i+1}|}$$

$$\epsilon_2|_i = \max(\phi_{i-1}, \phi_i, \phi_{i+1})$$

and $\alpha^* = \max(0, \alpha - c_2\epsilon_2)$. The details of our implementation are given in subsection II.B.1.

Figure 7 shows the pressure coefficient as a function of the c_2 constant in equation 1 on a coarse mesh (128×48). The oscillations close to shock, which are typical of the high-order dissipation operators, seem to be effectively reduced, although this effect is more pronounced for the 2nd order scheme. The same behaviour is observed of the fine grid, fig. 5. Unfortunately, for this configuration, with the grid described before, we were not able to obtain a grid converged solution with the required precision, nor with nor without shock capturing (see tab. 2 and 3). We are therefore unable to compute the errors.

II.B.1. Implementation details of the shock capturing scheme

Some tuning in order to make it work with our Newton-Krylov solver:

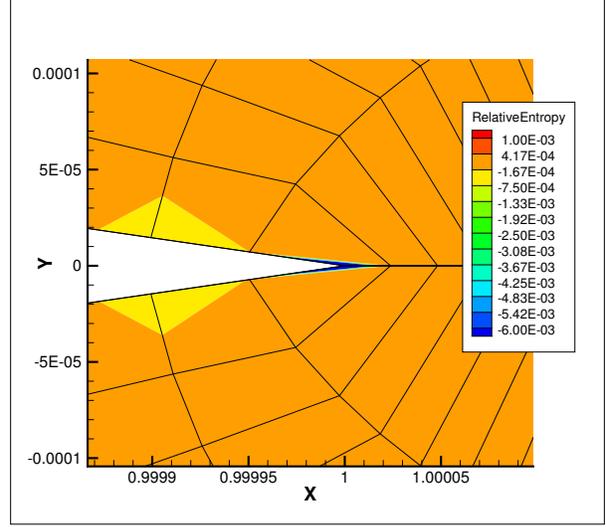


Figure 3. Entropy error contours, zoom close to trailing edge. 5th order scheme, 577×513 grid. Subsonic inviscid.

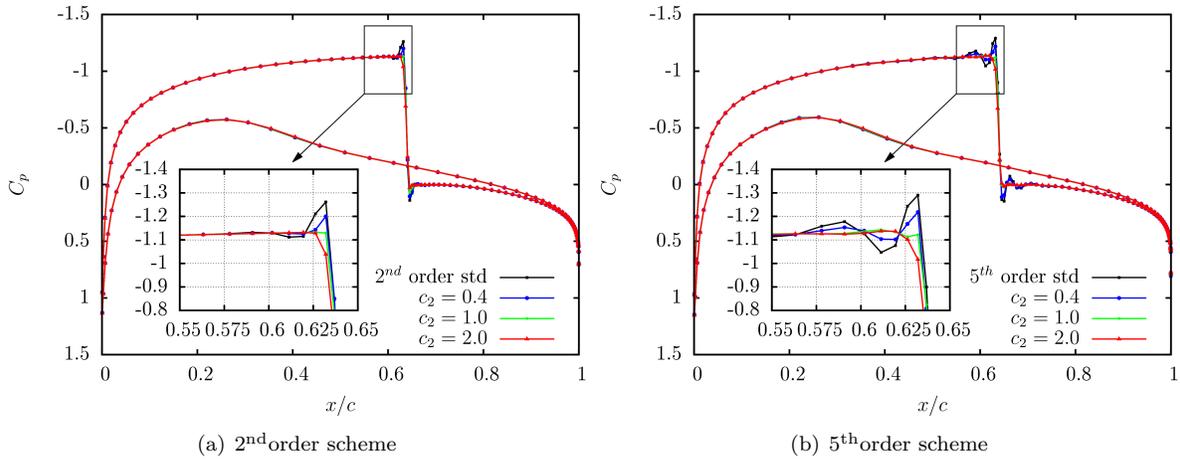


Figure 4. Transonic inviscid: Pressure coefficient profiles obtained with different values of the artificial dissipation constant c_2 (equation 1), coarse grid. The black line corresponds to $c_2 = 0$, i.e. no shock capturing.

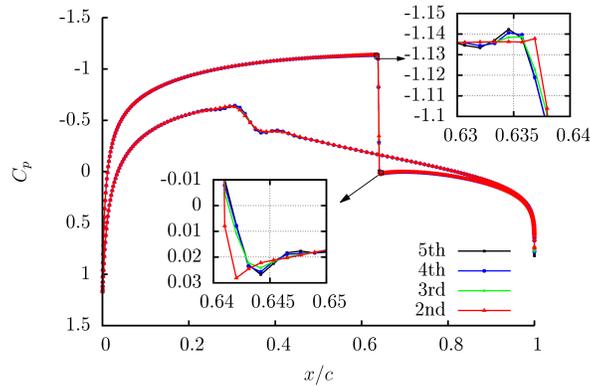


Figure 5. Transonic inviscid: Pressure coefficient profiles on the fine grid with $c_2 = 2.0$ (equation 1), 2nd to 5th order scheme.

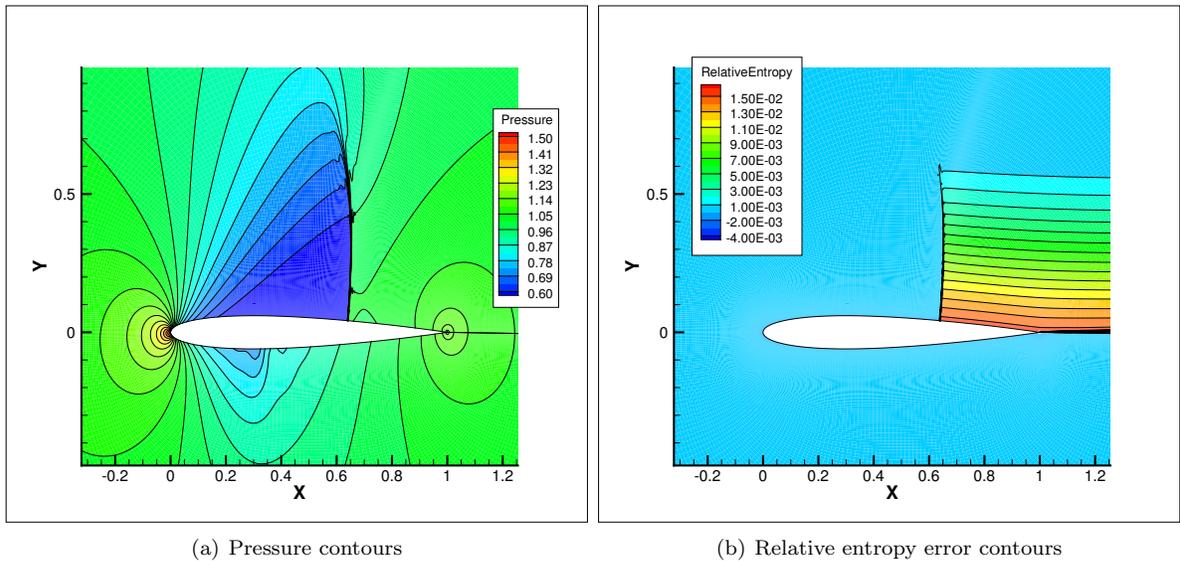


Figure 6. Transonic inviscid: fine grid results *with* shock capturing ($c_2 = 2.0$), 5th order scheme.

Table 2. 5th order results for the transonic inviscid configuration, without shock capturing.

C_l	C_d	h	work units
3.51991e-01	2.26498e-02	7.31175e-03	11.7
3.51682e-01	2.26161e-02	3.66931e-03	97.5
3.51607e-01	2.26216e-02	1.83803e-03	1425.6

Table 3. 5th order results for the transonic inviscid configuration, with shock capturing.

C_l	C_d	h	work units
3.48051e-01	2.28588e-02	7.31175e-03	31.8
3.51929e-01	2.26727e-02	3.66931e-03	212.3
3.51717e-01	2.26320e-02	1.83803e-03	1742.3

- ϕ_i is slightly different, it is computed as

$$\phi_i = \left[\frac{|p_{i-1} - 2p_i + p_{i+1}|}{|p_{i-1} + 2p_i + p_{i+1} + p_{lim}|} \right]^{1.1} \quad \text{with } p_{lim} = 0.001p_\infty$$

- $\epsilon_2 = \max(\epsilon_2, 0.25)$
- α^* is blended close to zero to avoid the clipping of the *max* function.

$$\alpha^* = \begin{cases} \alpha - c_2\epsilon_2 & \text{if } (\alpha - c_2\epsilon_2) \geq a \\ \frac{a^2 - b(\alpha - c_2\epsilon_2)}{(-b + 2a) - (\alpha - c_2\epsilon_2)} & \text{if } (\alpha - c_2\epsilon_2) < a \end{cases}$$

with $a = 1.e - 5$ and $b = 1.e - 12$. a is where the blending begins and b is the asymptotic value when $c_2\epsilon_2 \rightarrow \infty$.

- ϵ_2 is frozen after 2 non-linear iterations, and the whole non-linear system is solved twice to machine precision.

The last modification turned out to be the most important one for stability: freezing the scheme was the only way to obtain a converged solution.

II.C. Subsonic viscous

This configuration was assumed to have a steady-state solution which has been sought using both a (implicit) Newton-Krylov and a (explicit) Runge-Kutta solver with local time stepping. However, with the sharp trailing edge geometry, we found that only the former gave a steady state solution (shown in fig. 7(a)) while the latter was not able to find this solution; in fact the explicit Runge-Kutta solver did not converge. We also computed a time-accurate solution using a third order ESDIRK scheme: we found a steady-state solution with the requested accuracy of $1.e - 6$, albeit after very long integration time (> 60 convective times) before which vortex shedding was observed, fig. 7(b). With a rounded trailing edge, all algorithms converged to a steady-state solution.

Unfortunately, for this configuration too, lift and drag do not exhibit a monotonic convergence with the 5th order scheme. The the only exception is the lift with the sharp trailing edge. However, we did obtain a 'converged' value with an accuracy of $1.e - 6$ for both lift and drag on both geometries, as shown in tables 4 and 5.

A. Background information for the SBP-SAT scheme

As is well-known, stability of a numerical scheme is a key property for a robust and accurate numerical solution. Proving stability for high-order finite-difference schemes on bounded domains is a highly non-trivial task. One successful way to obtain stability proofs is to employ so-called Summation-by-Parts (SBP)

Table 4. 5th order results for the subsonic viscous configuration, rounded trailing edge.

C_l	C_d	h	work units
1.83387298e-02	5.53166144e-02	7.31175e-03	15.9
1.83515984e-02	5.53176140e-02	3.66931e-03	64.2
1.83509935e-02	5.53175559e-02	1.83803e-03	303.6

Table 5. 5th order results for the subsonic viscous configuration, sharp trailing edge.

C_l	C_d	h	work units
1.82572477e-02	5.53165933e-02	7.31175e-03	16.0
1.82714563e-02	5.53167410e-02	3.66931e-03	61.9
1.82721535e-02	5.53166991e-02	1.83803e-03	321.9

schemes with Simultaneous Approximation Terms (SAT) for imposing boundary conditions. With a simple example, we will briefly describe how stability proofs can be obtained.

Consider the scalar advection equation,

$$\begin{aligned} u_t + au_x &= 0, & 0 < x < 1, & \quad 0 < t \leq T \\ a^+ u(0, t) &= a^+ g_l(t) \\ a^- u(1, t) &= a^- g_r(t) \end{aligned} \quad (2)$$

where $a^+ = \max(a, 0)$ and $a^- = \min(a, 0)$. Furthermore, we augment the equation with initial data $u(x, 0) = f(x)$, bounded in L^2 . To demonstrate well-posedness, we employ the energy method.

$$\begin{aligned} \|u\|_t^2 + a \int_0^1 uu_x dx &= 0 \\ \|u\|_t^2 \leq au^2(0, t) - au^2(1, t) &\leq a^+ g_l(t)^2 - a^- g_r(t)^2 \end{aligned} \quad (3)$$

Integrating in time gives the bound

$$\|u(\cdot, T)\| \leq \|f\| + a^+ \int_0^T g_l(t)^2 dt - a^- \int_0^T g_r(t)^2 dt. \quad (4)$$

For linear PDEs, such a bound is sufficient to prove well-posedness.

Next, we turn to the SBP-SAT semi-discretization of (2). To this end, we introduce the computational grid, $x_i = ih$, $i \in \{0, 1, 2, \dots, N\}$ and $h > 0$ is the grid spacing. For the moment, we keep time continuous. With each grid point x_i , we associate a value $v_i(t)$, and define a grid function $v(t) = (v_0, v_1, v_2, \dots)^T$. The SBP difference operator, D is a matrix with the following properties: $D = P^{-1}Q$ where P and Q are two matrices; $P = P^T > 0$ and $Q + Q^T = B = \text{diag}(-1, 0, \dots, 0, 1)$. The matrix P can be used to define a weighted l^2 equivalent norm as $\|v\|^2 = v^T P v$. We will also need the vectors $e_0 = (1, 0, 0, \dots, 0)^T$ and $e_N = (0, \dots, 0, 1)^T$.

Let w denote a smooth function and define a grid function $\bar{w} = (w(x_0), \dots, w(x_N))^T$ and $\bar{w}_x = (w_x(x_0), \dots, w_x(x_N))^T$. It turns out that the SBP property precludes the accuracy of D to be uniform in space. We have

$$D\bar{w} = \bar{w}_x + \bar{T}$$

where \bar{T} is the truncation error. In general, it takes the form,

$$\bar{T}^T = (\mathcal{O}(h^s), \dots, \mathcal{O}(h^s), \mathcal{O}(h^p), \dots, \mathcal{O}(h^p), \mathcal{O}(h^s), \dots, \mathcal{O}(h^s)). \quad (5)$$

where $s < p$ and the lower accuracy is confined to a few (finite) number of points close to the boundary. SBP operators exist with various orders of accuracy, [4]. In particular, if P is a diagonal matrix, there are SBP operators with p even and $p \leq 8$, and $s = p/2$. If P is allowed to have off-diagonal elements for a few points near the boundary $s = p - 1$ can be achieved.

Using the SBP operators, we now define a semi-discrete scheme for (2).

$$v_t + aDv = \sigma_l a^+ P^{-1} e_0(v_0 - g_l(t)) + a^- \sigma_r P^{-1} e_N(v_N - g_r(t))$$

The right-hand side are the SAT:s, which impose the boundary conditions weakly. (Originally proposed in [5].) $\sigma_{l,r}$ are two scalar parameters, to be determined by the stability analysis. Multiplying by $2v^T P$, we obtain

$$\|v\|_t^2 - a(v_0^2 - v_N^2) = 2\sigma_l a^+ v_0(v_0 - g_l(t)) + 2a^- \sigma_r v_N(v_N - g_r(t)) \quad (6)$$

For stability, it is sufficient to obtain a bound with $g_{l,r} = 0$. In that case, it is easy to see that we must require $\sigma_l \leq -1/2$ and $\sigma_r \geq 1/2$ to obtain a bounded growth of $\|v\|$. More generally, allowing boundary data to be inhomogeneous when deriving a bound leads to *strong stability*. (See [6]. The benefit of proving strong stability as opposed to stability is that less regularity in the boundary data is required.) For strong stability, it can be shown that $\sigma_{l,r}$ must satisfy $\sigma_l < -1/2$ and $\sigma_r < 1/2$, i.e., strict inequalities. As an example, the choice $\sigma_l = -1, \sigma_r = 1$ leads to

$$\|v\|_t^2 - a(v_0^2 - v_N^2) = -2a^+ v_0(v_0 - g_l(t)) + 2a^- v_N(v_N - g_r(t))$$

or

$$\|v\|_t^2 \leq -a^+(v_0 - g)^2 + a^+ g_l(t)^2 + a^-(v_N - g_r(t))^2 - a^- g^2 \quad (7)$$

If $v_0 = g_l, v_N = g_r$, (7) is the same as (3), but this is not the case and the additional terms add a small damping to the boundary. Upon integration of (7) in time, an estimate corresponding to (4) is obtained. We also remark that the SAT terms are accurate as they do not contribute to a truncation error in the scheme. Furthermore, semi-discrete stability guarantees stability of the fully discrete problem obtained by employing Runge-Kutta schemes in time, [7].

The above example, demonstrates the general procedure for obtaining energy estimates for an SBP-SAT scheme. Naturally, for systems of PDEs, in 3-D with stretched and curvilinear multi-block grids, and with additional parabolic terms, the algebra for proving stability becomes more involved. However, the resulting schemes are still fairly straightforward to use. For the linearized Euler and Navier-Stokes equations, semi-discrete energy estimates have been derived. (See [8–10] and references therein.) Different boundary types, including far-field, walls and grid block interfaces are included in the theory. For flows with smooth solutions, linear stability implies convergence as the grid size vanishes. (See [11].)

B. Code description

Both a general code and specialized codes for some of the test cases (used in the 1st and 2nd high order workshop, see [12]) are available. The general code is a 3D code that can handle multiblock grids and can run on (massively) parallel platforms. For load balancing reasons the blocks are split during runtime in an arbitrary number of sub-blocks with a halo treatment of the newly created interfaces, such that the results are identical to the sequential algorithm.

The specialized codes assume a single block 2D grid and do not have parallel capabilities, hence they are relatively easy to modify for testing purposes. Due to the fact that these codes can only be used for one specific test case and the fact that the general purpose code can only handle 3D problems, the efficiency of the specialized codes is quite a bit higher than the general purpose code.

The discretization schemes used are finite difference SBP-SAT schemes, see section A, of order 2 to 5. Thanks to the energy stability property of these schemes no or a significantly reduced amount of artificial dissipation is needed compared to schemes which do not possess this (or a similar) property. This leads to a higher accuracy of the numerical solutions.

For the steady test cases the set of nonlinear algebraic equations is solved using the nonlinear solver library of PETSc [13]. This library requires the Jacobian matrix of the spatial residual, which is computed via dual numbers [14] and appropriate coloring of the vertices of the grid, for which the PETSc routines are used. Initial guesses are obtained via grid sequencing, where appropriate. The solution of the linear systems needed by PETSc's nonlinear solution algorithm is obtained by Block ILU preconditioned GMRES.

Implicit time integration schemes of the ESDIRK type [15] are available, for which the resulting nonlinear systems are solved using a slightly adapted version of the steady state algorithm explained above. However, for the unsteady test cases considered, the Euler vortex and the Taylor-Green vortex, the time steps needed for accuracy are relatively small compared to the stability limit of explicit time integration schemes and therefore the explicit schemes are better suited for these cases. The available explicit schemes are the classical 4th order Runge Kutta scheme (RK4, [16]) and TVD Runge Kutta schemes up till 3rd order [17]. As the maximum CFL number of the RK4 scheme is significantly higher than the CFL number of the TVD Runge Kutta schemes, the RK4 scheme is used for the unsteady test cases mentioned above.

For the post processing standard commercially available software, such as Tecplot, and open-source software, such as Gnuplot, are used. Grid adaption has not been carried out.

C. Machines description

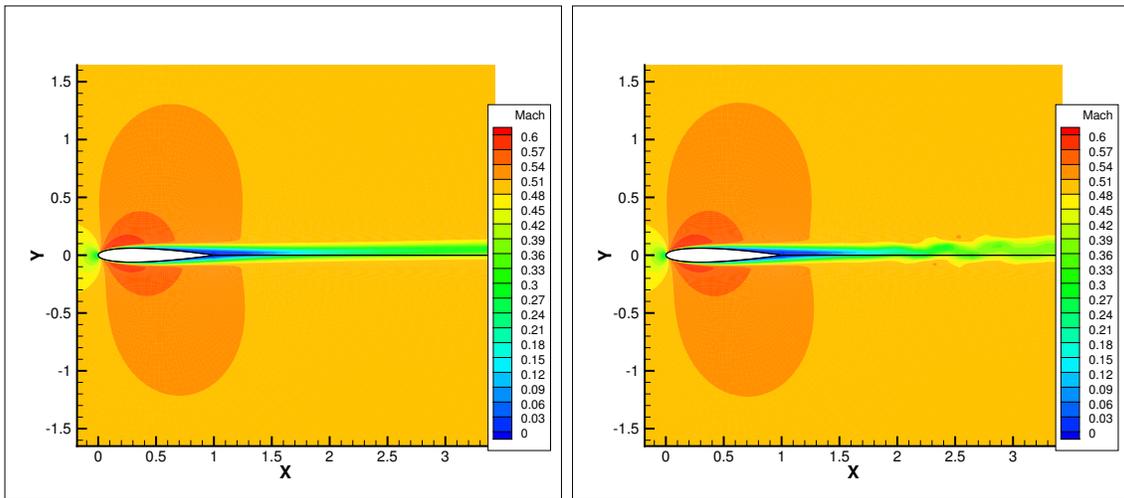
The results for the easy test cases have been obtained on a Linux work station running Ubuntu 10.04 with an Intel i7-2600 CPU running at 3.4 GHz, with 8 Mb of cache. The machine contains 16 Gb of RAM memory with an equivalent amount of swap. Running the Taubench on this machine led to a CPU time of 5.59 seconds (average over 4 runs).

The difficult test cases were run on up to 512 processors on the LISA machine of SARA, the Dutch Supercomputing Center and Hexagon, the Cray XE6 machine of the University of Bergen. Running the Taubench on these machines led to a CPU time of 10.3 and 10.8 seconds respectively (average over 4 runs).

References

- ¹ “Pointwise User Manual,” Tech. rep., www.pointwise.com, 2011.
- ² Mattsson, K., Svärd, M., and Nordström, J., “Stable and Accurate Artificial Dissipation,” *Journal of Scientific Computing*, Vol. 21(1), August 2004, pp. 57–79.
- ³ Jameson, A., Schmidt, W., and Turkel, E., “Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes,” *AIAA Paper 81-1259*, 1981.
- ⁴ Strand, B., “Summation by Parts for Finite Difference Approximations for d/dx ,” *J. Comput. Phys.*, Vol. 110, 1994.
- ⁵ Carpenter, M. H., Gottlieb, D., and Abarbanel, S., “Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes,” *J. Comput. Phys.*, Vol. 111(2), 1994.
- ⁶ Gustafsson, B., Kreiss, H.-O., and Oliger, J., *Time dependent problems and difference methods*, John Wiley & Sons, Inc., 1995.
- ⁷ Kreiss, H.-O. and Wu, L., “On the stability definition of difference approximations for the initial boundary value problem,” *Applied Numerical Mathematics*, Vol. 12, 1993, pp. 213–227.
- ⁸ Svärd, M., Carpenter, M., and Nordström, J., “A stable high-order finite difference scheme for the compressible Navier-Stokes equations, far-field boundary conditions,” *Journal of Computational Physics*, Vol. 225, 2007, pp. 1020–1038.
- ⁹ Svärd, M. and Nordström, J., “A stable high-order finite difference scheme for the compressible Navier-Stokes equations, no-slip wall boundary conditions,” *J. Comput. Phys.*, Vol. 227, 2008, pp. 4805–4824.
- ¹⁰ Nordström, J., Gong, J., van der Weide, E., and Svärd, M., “A stable and conservative high order multi-block method for the compressible Navier-Stokes equations,” *J. Comput. Phys.*, Vol. 228, 2009, pp. 9020–9035.
- ¹¹ Strang, G., “Accurate partial difference methods II. Non-linear problems,” *Num. Math.*, Vol. 6, 1964, pp. 37–46.

- ¹² van der Weide, E., Giangaspero, G., and Svärd, M., “Efficiency Benchmarking of an Energy Stable High-Order Finite Difference Discretization,” *AIAA Journal*, 2015, doi: 10.2514/1.J053500.
- ¹³ Balay, S., Brown, J., Buschelman, K., Eijkhout, V., Gropp, W., Kaushik, D., Knepley, M., McInnes, L. C., Smith, B., and H.Zhang, “PETSc Users Manual, Revision 3.2,” Tech. rep., Argonne National Laboratory, 2011.
- ¹⁴ Fike, J. A., Jongsma, S., Alonso, J., and v.d. Weide, E., “Optimization with Gradient and Hessian Information Calculated Using Hyper-Dual Numbers,” *AIAA paper 2011-3807*, 2011.
- ¹⁵ Kennedy, C. and Carpenter, M. H., “Additive Runge-Kutta schemes for convection-diffusion-reaction equations,” *Applied Numerical Mathematics*, Vol. 44, 2003, pp. 139–181.
- ¹⁶ Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 3rd ed., 2007.
- ¹⁷ Gottlieb, S., Shu, C. W., and Tadmor, E., “High order time discretizations with strong stability property,” *SIAM Review*, Vol. 43, 2001, pp. 89 – 112.



(a) Steady state solution obtain with the Newton-Krylov solver. (b) Instantaneous solution obtained with the third order ESDIRK scheme before the shedding is damped out.

Figure 7. Mach contours, 5th order scheme, fine grid, sharp trailing edge.